COMPUTATION OF DELAYS IN ACYCLICAL DISTRIBUTED

DECISIONMAKING ORGANIZATIONS*

by

Victoria Yu-yu Jin**

Alexander H. Levis**

ABSTRACT

An algorithm for computing time delays in a distributed decisionmaking system is developed. Starting with a matrix representation of the organizational structure all possible information processing paths are scanned and the time delay associated with each one is computed. When the decision strategies are known, the expected delay of the overall system can be obtained.

1. REPORT DATE **AUG 1985**	2. REPORT TYPE	3. DATES COVERED **00-08-1985 to 00-08-1985**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Computation of Delays in Acyclical Distributed Decisionmaking Organizations	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,Laboratory for Information and Decision Systems,77 Massachusetts Avenue,Cambridge,MA,02139-4307**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **20**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

1. INTRODUCTION

Decisionmakers in a distributed system have access to specified information and control specified resources. Usually, even in a simple organization, there is more than one path through which information can be processed. Decisionmakers can choose the path. There is no general rule for predicting which path will be chosen, because each individual decisionmaker (DM) has a different personality, different skills and reacts differently to different circumstances.

There are many measurements of performance of distributed decisionmaking (DDM) systems. One of the most important is the time interval from the moment a stimulus is received by a system to the moment a response is made. This time delay is one indicator of a system's ability to respond to events in a timely manner (see Cothier and Levis, 1985).

To evaluate the time delay in DDM systems, all possible information-processing paths must be identified and then the time delay associated with each path can be computed. A DDM system is often a large-scale system which contains many decisionmakers and decision support systems with complicated interconnections. For these systems, scanning all possible paths and computing time delays can become difficult or even impossible. An algorithm is required to solve the problem. Then, protocols that reduce the time delay in the operation can be designed, so that the effectiveness of the system can be improved [Jin, 1985].

In this paper, such an algorithm for computing time delay of DDM systems is developed, which scans all paths and computes conditional probabilities and time delays associated with these paths. From these results, a tree can be established to show all possible paths explicitly, the probability that each possible path occurs is easily calculated, and the expected delay of the overall system can be obtained.

The algorithm is developed by using the Petri Net representation of decisionmaking organizations [Tabak and Levis, 1985] which shows explicitly

2

the interactions between DMs and the sequence of operations in the system. Figure 1 shows the Petri Net representation of a two decisionmaker system; details of the procedure for constructing the Petri Net can be found in Tabak and Levis (1985).
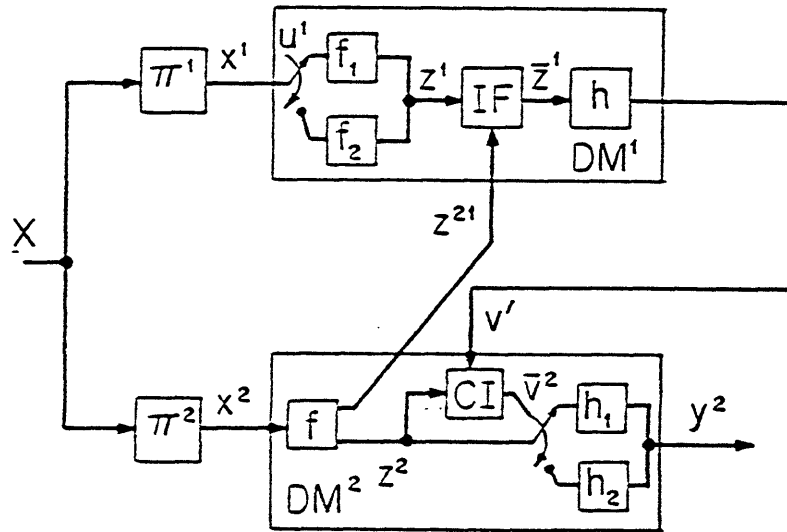
(a) A block diagram representation
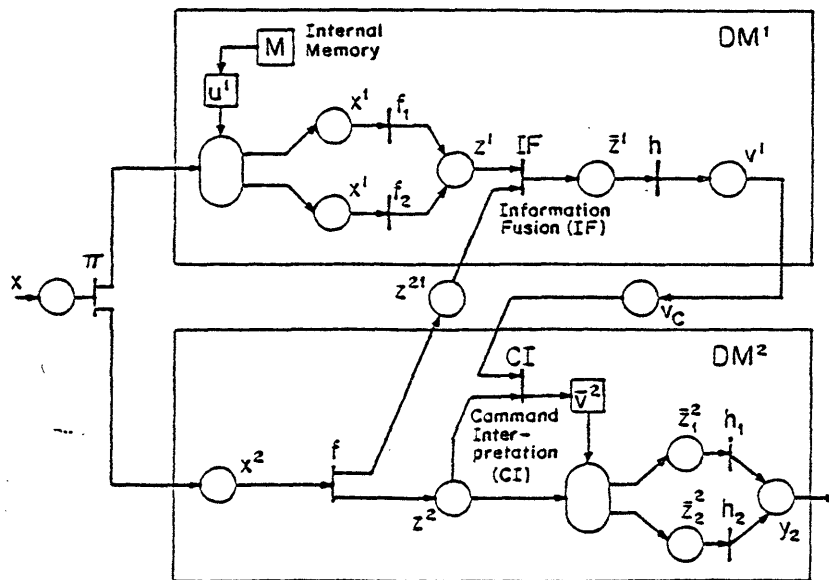
(b) An equivalent Petri Net representation

Figure 1. An Example of a Two Decisionmaker System

2. COMPUTATION OF TIME DELAYS

2.1 DEVELOPMENT OF AN INTERCONNECTION MATRIX

The information contained in a Petri Net can be summarized in the system matrix A_s, which is relatively complicated because of its compact form. Therefore, it is not convenient for scanning for all possible paths. An interconnection matrix, C_s, is needed that indicates the interconnections between DMs in an explicit way so that all possible paths can be found using a simple algorithm. The interconnection matrix is obtained by scanning the system matrix and storing the relevant interconnection information in a new format. The interconnection matrix indicates whether the components are connected (+1, -1) or not connected (0); and, if connected, how they are connected.

The elements of the interconnection matrix, C_s, are defined as follows:

1. C_s has a dimension of m x n, where m is the total number of arcs (or links) in a Petri Net and n is the total number of transitions.

2. An element of C_s, $\{c_{ij}\}$, gives the connection status of the j-th transition to the i-th arc;

3. The element c_{ij} can take the values of -1, 0, +1:

$$c_{ij} = \begin{cases} -1, & \text{when there is an output from the j-th transition to the i-th link;} \\ 0, & \text{when there is no connection between the j-th transition and the i-th link;} \\ +1, & \text{when there is an input from the i-th link to the j-th transition.} \end{cases}$$

As an example, Table 1 shows the system matrix A_s for the two-decisionmaker organization of Figure 1. In the system matrix, each decisionmaker is considered as a subsystem. The information source (AIN) and the response sink (AOU) are also subsystems. Therefore, the total number of subsystems in a DDM system with n DMs is n+2. Each transition of

4

a Petri Net is modeled as a column in a submatrix, which contains the input and output information for the transition.

Table 1. The System Matrix of the Two DM System

$$
A_S = \begin{array}{c}
\text{AIN} \\
\\
\\
\\
\\
\\
\\
\\
\\
\text{A1} \\
\\
\\
\\
\\
\text{A2} \\
\\
\\
\\
\\
\text{AOU} \\
\\
\\
\\
\\
\end{array}
\left[
\begin{array}{cccc}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1111 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1.12 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
3333 & 3333 & 2 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 2 \\
0 & 0 & 0 & 2 \\
1 & 1 & 1 & 1 \\
3333 & 1 & 0 & 0 \\
1 & 4 & 0 & 0 \\
1 & 0 & 4444 & 4444 \\
3 & 0.12 & 1 & 1 \\
1 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
3.12 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\right]
$$

The elements of each column indicate the origins of the inputs to that transition and the destination of its outputs. For example, column 1 in AIN (Table 1) indicates that the delay of this transition is one unit; the second element shows that there are no inputs and hence the third, which would otherwise indicate the source, is also zero. The fourth entry, 1111, is a code indicating multiple outputs. There are 2 outputs (fifth entry); the first output is routed to decisionmaker #1 (1 in position 6), to his first transition which is a two-way switch (the seventh entry, 1.12). The other output goes to decisionmaker #2 and, specifically, to his first transition (eight and ninth entries). For details, see Tabak and Levis (1985). The corresponding interconnection matrix is given in Table 2.

Table 2. The Interconnection Matrix for the Two DM System

$$C_S = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix}$$

After C_S is established, it is necessary to check whether the sum in each row is zero. If it is not, there must be an error because each row of C_S stands for only one link which connects two vertices in a certain direction.   Therefore, there must be a −1 to indicate that link is an output of one of the vertices and a +1 to indicate that it is an input of the other vertex.

2.2 SCANNING ALL POSSIBLE PATHS

The scanning problem is formulated as finding all possible paths from the vertex that represents the input source to the vertex that is the output sink.   The paths form a tree with the input source as the main root of the tree. Every path is a branch of the tree.

The Algorithm

Let $P(m,z)$ be the z-th subpath ending at the m-th vertex and D_{mz} be the time delay associated with this subpath. The elements of C_S are partitioned into four subsets: S_1 and $\bar{S}_1 = C_S - S_1$, S_2 and $\bar{S}_2 = C_S - S_2$ with $S_1 = \{1\}$ and $S_2 = \{-1\}$.

The elements of C_S have the following properties:

6

(1) If $c_{ij} = -1$ and $c_{ik} = 1$, vertices j and k are connected and V_j precedes V_k.

(2) If there are more than one (-1) in column j of C_s, vertex V_j is a root or a subroot.

(3) If there are n $(+1)$ in column j, then n paths converge into the same path after they reach vertex V_j.

Scanning is done backwards, that is, it starts from the last vertex of the output sink, V_i, in which there are usually more than one input. The first positive one $(+1)$ in the i-th column of C_s is the first input to V_i, and it is processed first. The processing stops when a multi-input vertex, V_j, is found, i.e., there are several paths converging into transition V_j. To avoid iterative computation, V_j is stored as a subroot and is marked as the end of some subpaths. Then scanning goes back to the second input of V_i. The previous procedure is repeated until a new convergent vertex, say, V_k, is found. After all inputs of V_i are processed, the same procedure is repeated for all the subroots. When the subpaths of the last subroot end with V_1, which is the first transition of the source, scanning is completed.

After all subpaths are found, they are assembled into paths by matching the last vertex V_k in the subpath $P(i,j)$ to the first vertex V_k in $P(k,z)$. When the last vertex of a subpath is V_1, a path is completed.

The algorithm depends on the following rule.

Let $c_{ik} \in S_1$ and $c_{hj} \in S_2$, i.e., $c_{ik} = +1$ and $c_{hj} = -1$.

If $i = h$

then there is a path from V_j to V_k, i.e.,

$$P(j,z) = V_j \rightarrow V_k$$

and the delay associated with this path, D_{jz}, is

$$D_{jz} = D_j + D_k,$$

the sum of the delays of the two vertices in the path.

One important rule of the algorithm is that no loop is allowed in any path. If a vertex appears in one path more than once, scanning stops. An error message is given.

Consider the system in Fig. 1; its system matrix A_s and interconection matrix C_s were given in Tables 1 and 2. There are 10 vertices and 13 edges. All paths are scanned using the algorithm. Related subpaths are then joined together to complete possible paths. The subpaths ending with the n-th vertex are connected to the subpaths starting on the n-th vertex to form intermediate paths. Table 3 lists all subpaths and intermediate paths for this example.

Often some possible paths are active simultaneously, i.e., in parallel. Therefore, to avoid confusion, intermediate paths are defined as follows:

An intermediate path is a single path which starts from the source vertex and ends at the sink vertex.

Then, a possible path can be represented by a "sequence" of intermediate paths. Figure 2 shows a tree which displays all possible paths as sequences of intermediate paths. Notice that even though intermediate paths are shown in sequence, this does not mean that they occur one after another, but instead, they may be simultaneous. The tree representation shows all possible paths explicitly.

8

Table 3. **All Subpaths and Intermediate Paths of 2 DM System**

Subpaths	Index of Vertex
P(10,1)	10, 9, 7
P(10,2)	10, 8, 7
P(7,1)	7, 4, 1
P(7,2)	7, 6, 5
P(5,1)	5, 4, 1
P(5,2)	5, 3, 1
P(5,3)	5, 2, 1

Intermediate Paths	Index of Vertex
P(1)	10, 9, 7, 4, 1
P(2)	10, 8, 7, 4, 1
P(3)	10, 9, 7, 6, 5, 4, 1
P(4)	10, 9, 8, 6, 5, 4, 1
P(5)	10, 9, 7, 6, 5, 2, 1
P(6)	10, 9, 7, 6, 5, 3, 1
P(7)	10, 8, 7, 6, 5, 3, 1
P(8)	10, 8, 7, 6, 5, 2, 1
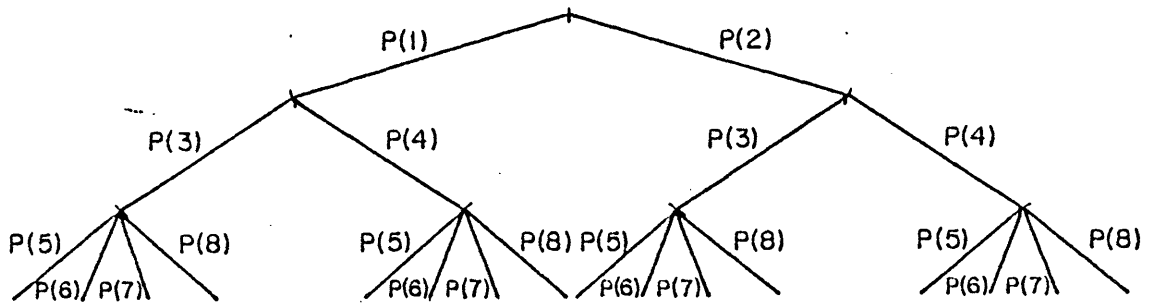
Figure 2. A Tree Showing All Possible Paths of the System

9

2.3 COMPUTATION OF TIME DELAY FOR ALL PATHS

To compute the delay for each path, the only calculation needed is to add the delays associated with the vertices which constitute a path. The algorithm for the computation of delays is:

(a) Let n be the number of transitions in the j-th subpath ending at vertex i, $P(i,j)$. Assume that a transition has a delay of t_k. Then the delay of $P(i,j)$ is

$$D_{ij} = \sum_{k=1}^{n} t_k \qquad (1)$$

(b) Let D_{i1}, D_{i2}, D_{i3} be time delays associated with three subpaths $P(i,1)$, $P(i,2)$, $P(i,3)$. Then the delay of all subpaths with the end vertex V_i is

$$D(i) = \max(D_{i1}, D_{i2}, D_{i3}) \qquad (2)$$

For instance, consider the example in Table 3.   There are four distinct subpaths ending at V_7:

P(7,1) = 7,4,1

P(7,2) = 7,6,5,2,1

P(7,3) = 7,6,5,3,1

P(7,4)-= 7,6,5,4,1

The associated delays are D_{71}, = 3, D_{72} = 5, D_{73} = 5, D_{74} = 5. Then the delay from V_1 to V_7 is

$$D(7) = \max(D_{71}, D_{72}, D_{73}, D_{74}) = 5 \qquad (3)$$

10

For an intermediate path containing n subpaths, the delay is the sum of delays associated with the n subpaths. For example, in Table 3, intermediate path P(1) contains subpaths P(10,1) and P(7,1). Then, the delay of P(1) is D(1)= 3 + 5 - 1 = 7 where 3 is the delay in subpath P(10,1); 5 is the delay in subpath P(7,1) which is calculated above; and 1 is subtracted because V_7 is counted twice.

For a possible path, because all the intermediate paths are simultaneous, the time delay is the maximal delay of the intermediate paths. For example, possible path 1 in Fig. 2 consists of intermediate path P(1), P(3), P(5). The maximal delay is 7, so the delay associated with path 1 is 7.

After all delays are computed, a shortest path with the minimal time delay and a longest path with maximal delay can be found. For analyzing overall system performance, it may be desired to compute the expected delay of the system.

2.4 EXPECTED DELAY OF A SYSTEM

To calculate expected delay, probabilities associated with each path need to be calculted first. Then the expected delay of a system can be computed.

Usually, for a system model, probabilities are given as conditional probabilities associated with each transition. If transition V_i has only one input from the previous transition, V_j, then the conditional probability $p(V_i/V_j)$ is 1. If V_i is a transition of a decision switch, a conditional probability $p(V_i/V_j) \leq 1$ will be assigned. For a n-way switch, the sum of n conditional probabilities should be equal to 1, that is

$$\sum_{i=1}^{n} p(V_i/V_j) = 1 \qquad (4)$$

11

The probability that information processing will follow a certain intermediate path X with n transitions is given by

$$p(X) = p(V_1) \prod_{i=1}^{n-1} p_i(Z/V) \qquad (5)$$

where X is the path number; Z is a transition which is on the path and V is the transition preceding Z. Table 4 shows the conditional probability matrix P_S for the example of section 2.1. Table 5 shows the probabilities associated with intermediate paths of System A.

Table 4. Probability Matrix P_S of 2 DM System

$$P_S = \begin{bmatrix}
0.0 & 0.6 & 0.4 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.7 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}$$

The expected delay can be calculated by the following equation:

$$E_r = \sum_{i=1}^{r} p_i D(i) \qquad (6)$$

where p_i and $D(i)$ are the probability and time delay associated to the i-th possible path; r is the total number of possible paths in a system.

Table 5. Conditional Probabilities Associated with Each Intermediate Path
          of 2 DM System

Intermediate Path	Conditional Probability
P(1)	1 * 1 * 0.7 * 1 = 0.7
P(2)	1 * 1 * 0.3 * 1 = 0.3
P(3)	1 * 1 * 1 * 1 * 0.7 * 1 = 0.7
P(4)	1 * 1 * 1 * 1 * 0.3 * 1 = 0.3
P(5)	0.6 * 1 * 1 * 1 * 0.7 * 1 = 0.42
P(6)	0.4 * 1 * 1 * 1 * 0.7 * 1 = 0.28
P(7)	0.4 * 1 * 1 * 1 * 0.3 * 1 = 0.12
P(8)	0.6 * 1 * 1 * 1 * 0.3 * 1 = 0.18

For example, in the 2 DM system, there are 16 possible paths (Figure 2). Path 1 has delay of 7 (Section 2.4) and probability of 0.206. Then the first term of E_n is 0.206 * 7. In this particular example, because the delay, D, of all possible paths is 7,

$$E_{16} = D \cdot \sum_{i=1}^{16} p_i = 7. \tag{7}$$

3. APPLICATION

The delays in two organizations, each one consisting of three decisionmakers, will be determined using the algorithm described in Section 2. The application is an abstracted and very simplified version of an air defense problem. In the parallel organization (Fig. 3), the airspace has been divided into three sectors, with each decisionmaker assigned to one sector. Each DM can observe and engage threats in his sector. However, threats may move between sectors; therefore, there is need for communication — information sharing — between decisionmakers with adjacent sectors. In the hierarchical organization (Fig. 4), the airspace
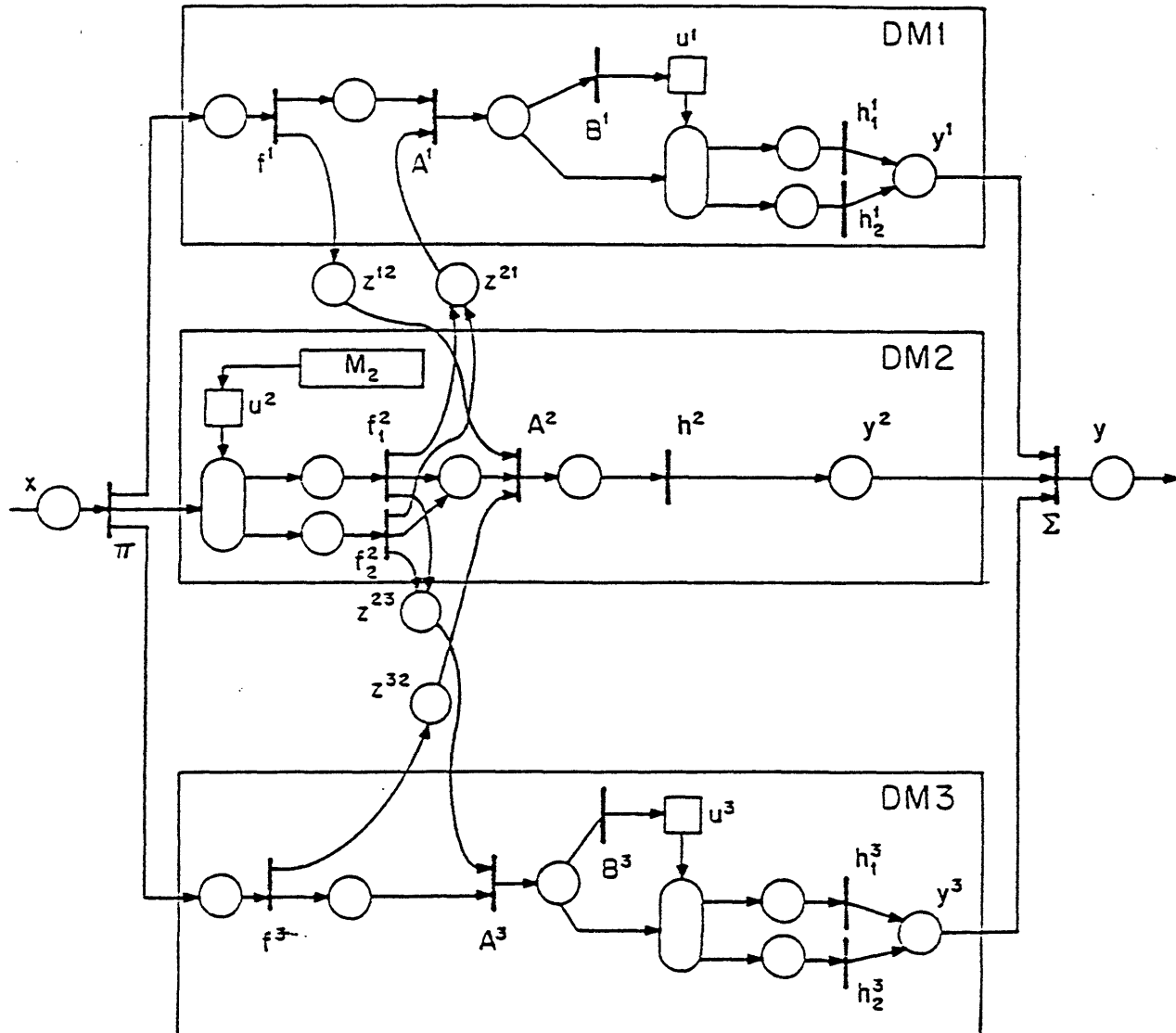
Figure 3. A Parallel Organization

is divided into two sectors, with each one assigned to a single DM. Since, the workload will be high for each DM, a central region is defined that stradles the two sectors. A supervisor is introduced who does not observe the airspace directly, but receives information about threats in the central region from the two DMs. He then processes the data and allocates threats in the central region (command inputs) to either one of the DMs depending on the trajectory of the threat.

Figure 4. A Hierarchical Organization

Parallel Organization

Using the algorithm interconnection matrix, all intermediate paths, the conditional probability and time delay associated with each intermediate path, and the expected delay of overall system are computed. The interconnection matrix and labels of transitions or vertices are shown in Table 6.

In the results, Table 7, intermediate paths are indicated by the sequence of indices of the vertices which represent transitions. The symbol "SW" denotes that the following vertex (transition) belongs to a decision switch. The symbol "/ /" indicates that the following subpath is parallel to other subpath(s).

Table 6. Interconnection Matrix for Parallel Organization

```
*****************************************************************
INTERCONNECTION MATRIX :   COLUMN=TRANSITION   ROW=LINK
*****************************************************************


-1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
-1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
-1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
-1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
 0  -1   0   0   0   1   0   0   0   0   0   0   0   0   0   0
 0  -1   0   0   0   0   1   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0  -1   1   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0  -1   1   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0  -1   1   0   0   0   0   0   0
 0   0   0   0   0   0   0  -1   0   0   1   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   1  -1   0   0   0   0   0
 0   0  -1   0   0   0   1   0   0   0   0   0   0   0   0   0
 0   0  -1   0   0   0   0   0   0   0   0   1   0   0   0   0
 0   0   0  -1   0   0   1   0   0   0   0   0   0   0   0   0
 0   0   0  -1   0   0   0   0   0   0   0   1   0   0   0   0
 0   0   0  -1   0   1   0   0   0   0   0   0   0   0   0   0
 0   0  -1   0   0   1   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0  -1   0   0   0   0   0   0   1   0   0   0
 0   0   0   0   0   0   0   0   0   1   0   0  -1   0   0   0
 0   0   0   0  -1   1   0   0   0   0   0   0   0   0   0   0
 0   0   0   0  -1   0   0   0   0   0   0   1   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0  -1   0   1   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0  -1   1   0
 0   0   0   0   0   0   0   0   0   1   0   0   0   0  -1   0
 0   0   0   0   0   0   0   0   0   0   0   0   0  -1   0   1
 0   0   0   0   0   0   0   0   0   1   0   0   0   0   0  -1
```
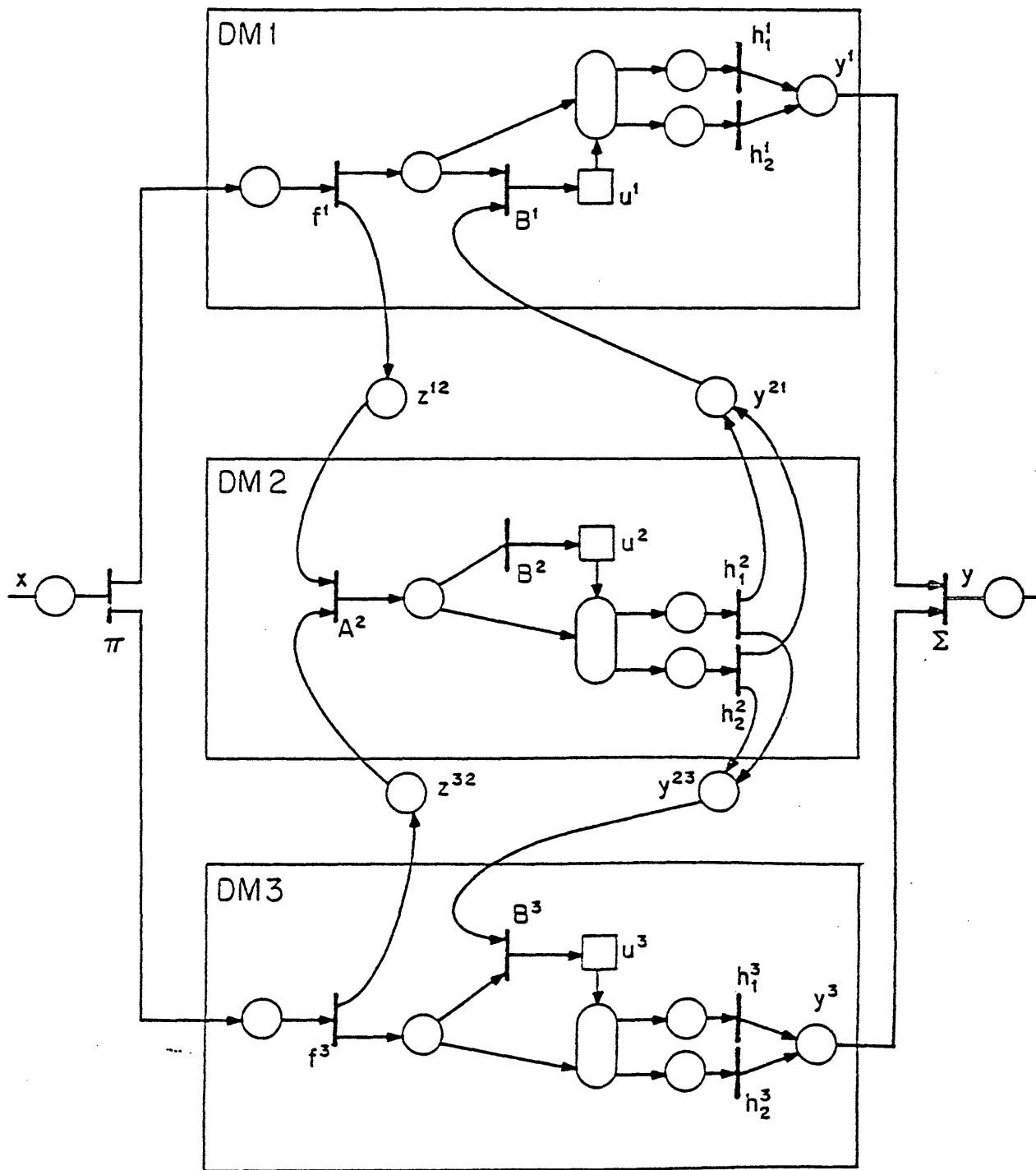
Table 7. Paths for Parallel Organization

```
==============================================================================
                                                    CONDITIONAL
                   INTERMEDIATE PATH                 PROBABILITY   DELAY
==============================================================================

P( 1)= 10<- SW-0 9<-0 8<-0 7<-//<-0 2<-0 1              0.300       6

P( 2)= 10<- SW-0 9<-0 8<-0 7<- SW-0 3<-0 1              0.120       6

P( 3)= 10<- SW-0 9<-0 8<-0 7<- SW-0 4<-0 1              0.180       6

P( 4)= 10<- SW- 11<-0 8<-0 7<-//<-0 2<-0 1              0.700       6

P( 5)= 10<- SW- 11<-0 8<-0 7<- SW-0 3<-0 1              0.280       6

P( 6)= 10<- SW- 11<-0 8<-0 7<- SW-0 4<-0 1              0.420       6

P( 7)= 10<-//<- 13<-0 6<-//<-0 2<-0 1                   1.000       5

P( 8)= 10<-//<- 13<-0 6<- SW-0 4<-0 1                   0.600       5

P( 9)= 10<-//<- 13<-0 6<- SW-0 3<-0 1                   0.400       5

P(10)= 10<-//<- 13<-0 6<-//<-0 5<-0 1                   1.000       5

P(11)= 10<- SW- 15<- 14<- 12<- SW-0 3<-0 1              0.200       6

P(12)= 10<- SW- 15<- 14<- 12<- SW-0 4<-0 1              0.300       6

P(13)= 10<- SW- 15<- 14<- 12<-//<-0 5<-0 1              0.500       6

P(14)= 10<- SW- 16<- 14<- 12<- SW-0 3<-0 1              0.200       6

P(15)= 10<- SW- 16<- 14<- 12<- SW-0 4<-0 1              0.300       6

P(16)= 10<- SW- 16<- 14<- 12<-//<-0 5<-0 1              0.500       6


==============================================================================
EXPECTED DELAY OF THE SYSTEM IS   6
==============================================================================
```

Table 7, shows that there are sixteen intermediate paths. Some of these paths are in parallel. For example, P(1) is parallel to both P(2) and P(3): they have the same subpath after V_7. Only decision switches create different paths. Because some intermediate paths occur simultaneously, that is, are parallel, consequently, the resulting tree has 128 possible paths.

A delay of unity in each transition is assumed during the computation. The expected delay of this system is 6 units.

Hierarchical Organization

The interconnection matrix and the transition labels for the hierarchical organization are shown in Table 8. All intermediate paths, their conditional probabilities and their delays are shown in Table 9. There are twenty intermediate paths. Each path has a delay of eight units, because for a set of parallel paths the delay of the set is the maximum path delay.

Table 8. Interconnection Matrix for Hierarchical Organization

```
*******************************************************************
INTERCONNECTION MATRIX :   COLUMN=TRANSITION   ROW=LINK
*******************************************************************
```

-1	1	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	1	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	1	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	-1	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	1	0	0	0	0
0	0	0	0	1	0	0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	1	-1	0	0
0	0	-1	1	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	-1	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	-1	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	-1

Table 9. Paths for Hierarchical Organization

INTERMEDIATE PATH	CONDITIONAL PROBABILITY	DELAY
P(1)=0 7<- SW-0 6<-0 5<-//<-0 2<-0 1	0.400	8
P(2)=0 7<- SW-0 6<-0 5<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.280	8
P(3)=0 7<- SW-0 6<-0 5<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.280	8
P(4)=0 7<- SW-0 6<-0 5<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.120	6
P(5)=0 7<- SW-0 6<-0 5<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.120	8
P(6)=0 7<- SW-0 8<-0 5<-//<-0 2<-0 1	0.600	8
P(7)=0 7<- SW-0 8<-0 5<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.420	8
P(8)=0 7<- SW-0 8<-0 5<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.420	8
P(9)=0 7<- SW-0 8<-0 5<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.180	8
P(10)=0 7<- SW-0 8<-0 5<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.180	8
P(11)=0 7<- SW- 13<- 11<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.350	8
P(12)=0 7<- SW- 13<- 11<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.350	8
P(13)=0 7<- SW- 13<- 11<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.150	8
P(14)=0 7<- SW- 13<- 11<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.150	8
P(15)=0 7<- SW- 13<- 11<-//<-0 3<-0 1	0.500	8
P(16)=0 7<- SW- 14<- 11<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.350	8
P(17)=0 7<- SW- 14<- 11<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.350	8
P(18)=0 7<- SW- 14<- 11<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.150	8
P(19)=0 7<- SW- 14<- 11<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.150	8
P(20)=0 7<- SW- 14<- 11<-//<-0 3<-0 1	0.500	8

EXPECTED DELAY OF THE SYSTEM IS   8

The resulting tree has 64 different possible paths, and the expected delay is 8 units.

4. CONCLUSIONS

An algorithm has been developed for computing time delays in DDM systems. From the system matrix, an interconnection matrix is created, which consolidates all the information about connections between the system's components. Scanning of the interconnection matrix results in a set of intermediate paths. Then, all possible paths can be constructed by concatenating intermediate paths. The time delay associated with each intermediate path is calculated by summing the delays of the transitions contained in that path. A possible path is composed of several intermediate paths which are active simultaneously. Then, the time delay associated with a possible path is the maximal delay of the intermediate paths contained in this possible path. After all possible paths and associated delays are found, the expected delay for the overall decisionmaking system can be calculated. The expected delay provides an indication of the speed of response of the system. The algorithm has been applied to compute the delays in parallel and hierarchical organizations.

5. REFERENCES

Cothier, P. H., and A. H. Levis, "Assessment of Timeliness in Command and COntrol," LIDS-P-1454, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, April 1985.

Jin, V. Y-Y, "Delays for Distributed Decisionmaking Organizations," MS Thesis, LIDS-TH-1459, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, May 1985.

Tabak, D., and A.H. Levis, "Petri Net Representation of Decision Models," to appear IEEE Trans. on Systems Man and Cybernetics, November 1985.